WebObjects Java Client

Session 707

WebObjects Java Client

Andreas Wendker
Manager, WebObjects Rapid Development

Introduction

- WebObjects gives you incredibly powerful database connectivity for desktop applications
- Java Client is a 3-tier application architecture
  - Client based on Swing, platform independent
  - Communication protocol: HTTP
    - Connectivity of an HTML-based application
    - Richer user interface
- Future: 2-tier application architecture

What You Will Learn

- How to use Interface Builder to create user interfaces in Swing
- What you need to know about deploying Java Client applications
- How the distribution layer works
- How to make Java Client applications secure

User Interfaces

- Direct to Java Client
  - Rule-based, code-free development
  - User interface dynamically from XML
  - Various customization techniques
- Interface Builder
  - User Interface created graphically
  - Use with Direct to Java Client

Why Interface Builder?

- EOF interfaces need rich connections
    - Associations with multiple aspects
- Interface Builder is the one tool available to us which provides the necessary connection facilities

Demo

Simple Java Client Application

Norbert Schatz
WebObjects Engineering

Working With Interface Files

- Interface Builder translates Cocoa user interfaces to Swing
- Interfaces loaded through EOInterfaceControllers:

```
protected void loadArchive ();
```

Working With Interface Files

- API on EOInterfaceController:

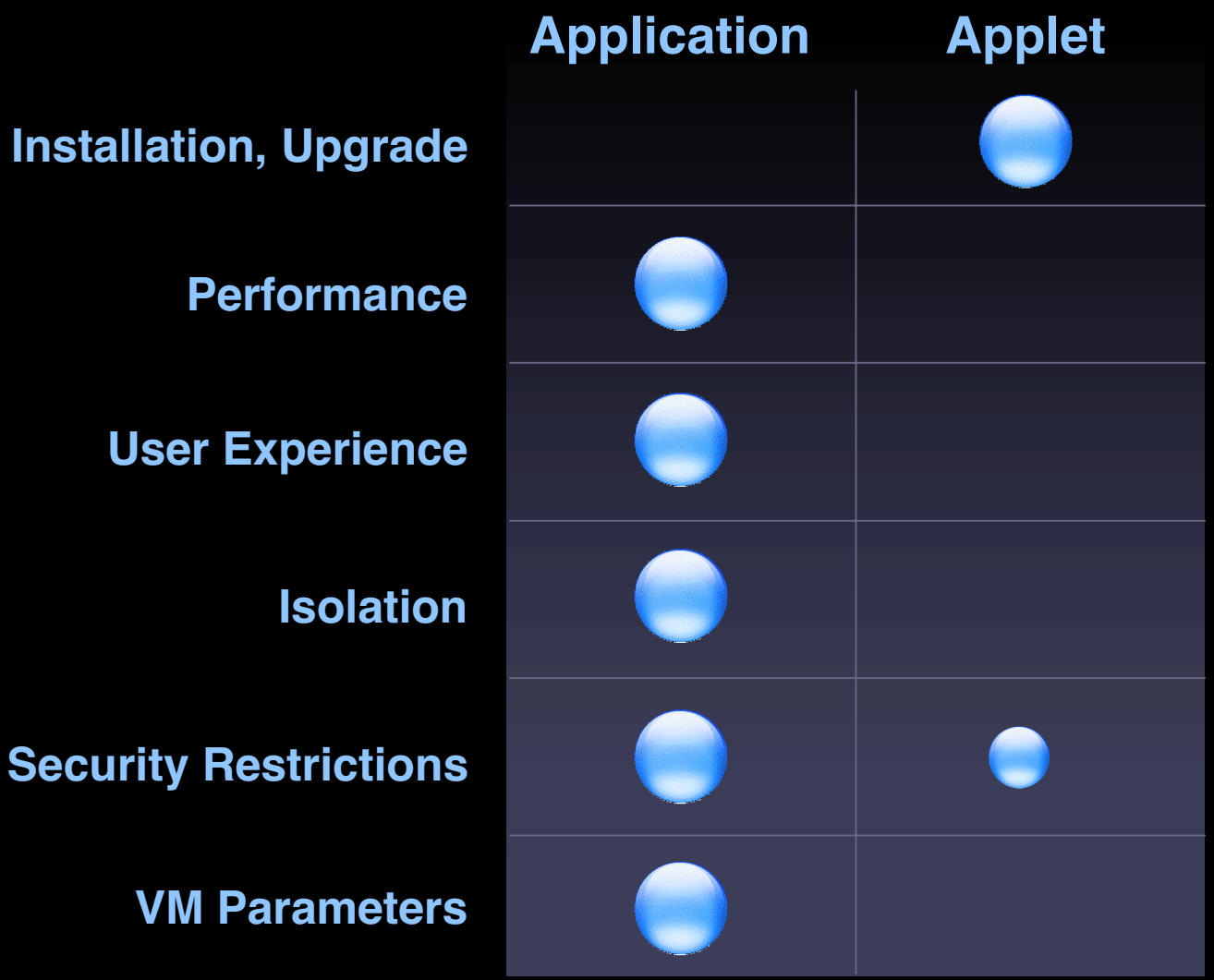
```
protected void controllerDidLoadArchive (
    NSDictionary namedObjects );

protected void controllerWillLoadArchive ();
```

- Programmatic completion
  - Outlets
  - Named objects API allows to plug-in other types of GUI editors

Client Deployment

- Client requirements: J2SE 1.3.1
  - Integrated into Mac OS X 10.1
- Currently, two client deployment options
  - Desktop applications (preinstalled)
  - Applets (downloaded every time)

Client Deployment Options



Web Start

- Part of J2SE 1.4, optional extension for J2SE 1.3
  - Integrated into Mac OS X 10.1 (Java Update)
- Launch mechanism
  - Entry page in HTML with a special hyperlink
  - Application executes outside the browser
- Solves installation/update problem
  - Client classes downloaded
  - Jar files cached
  - Automatic version check at startup

Web Start

- Security for each individual application
  - User grants permissions at startup
  - Signed jar files provide “better illusion” of security
- Web Start Manager
  - Start applications without visiting entry page
  - Create dockable desktop applications (Mac OS X)
  - Create shortcuts (Windows)

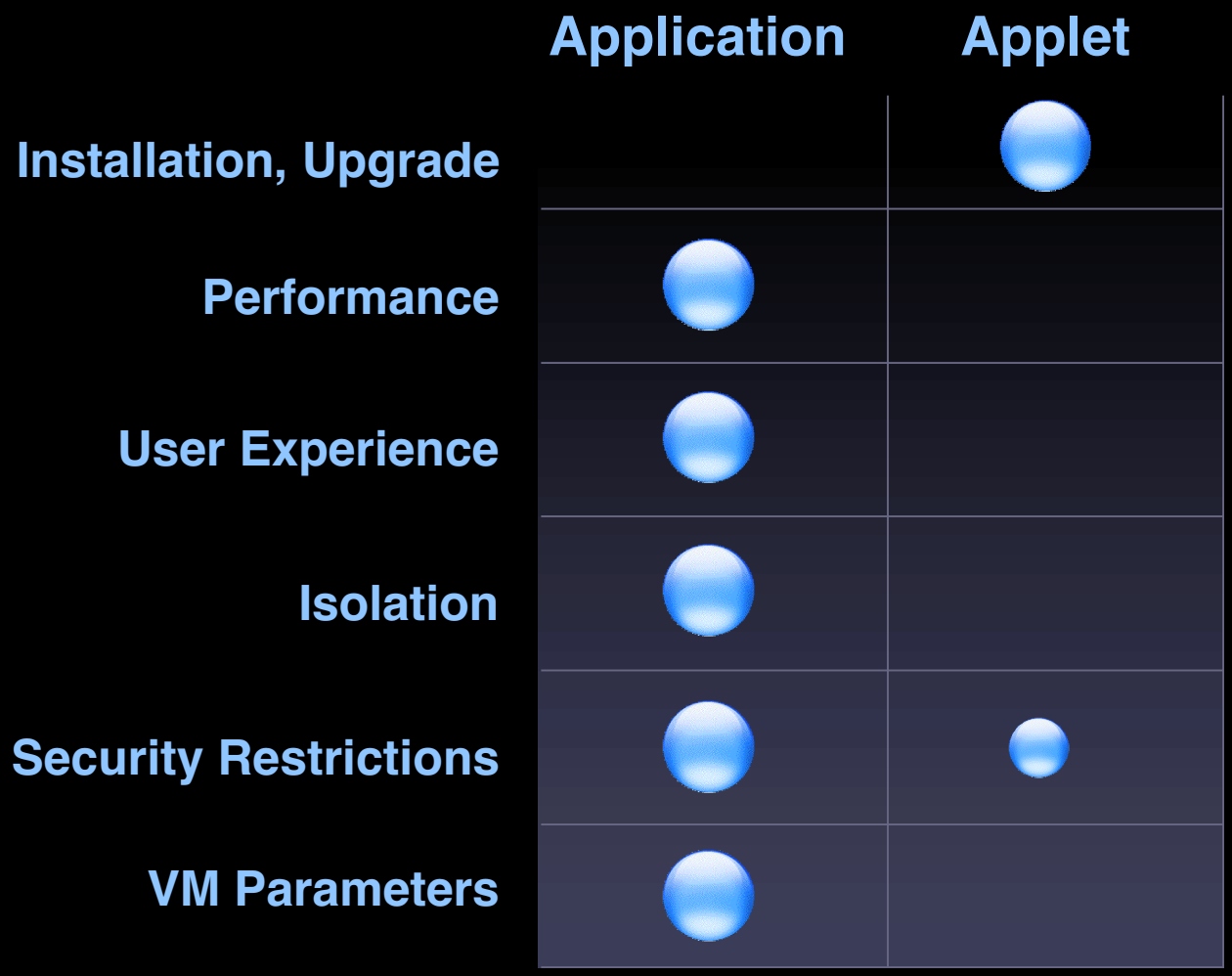
Web Start

- Client application information stored in JNLP file
  - Application name
  - Version
  - Description
  - Vendor
  - Homepage
  - Jar file resources

Web Start and WebObjects

- Deprecation of Applet support
- Tight integration of Web Start
  - Automatic generation of JNLP file at runtime
  - Project Builder templates will change to reflect the need for pretty entry pages

Client Deployment Options



Demo

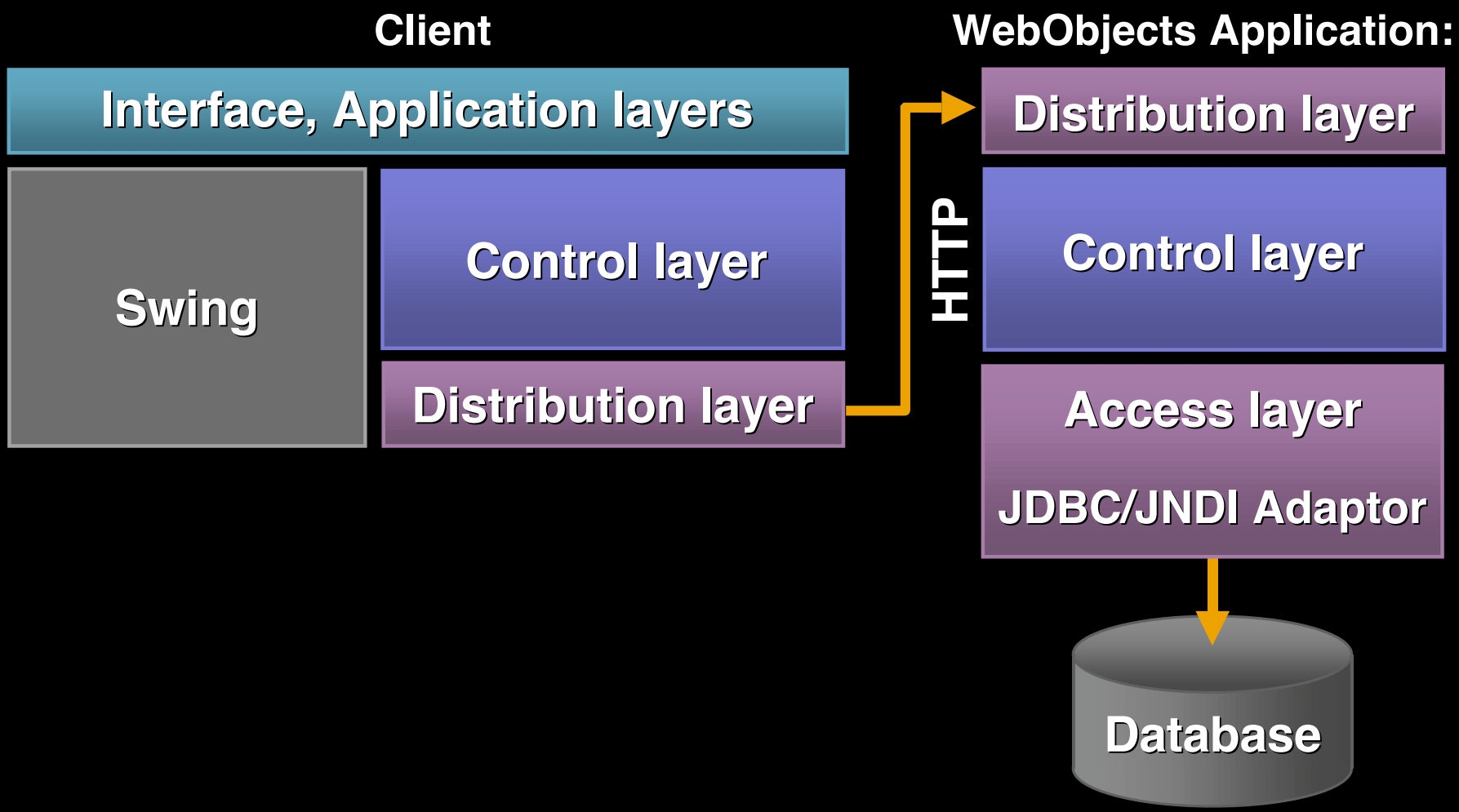
Java Client With Web Start

Norbert Schatz
WebObjects Engineer

Distribution Layer

- Copy distribution mechanism, no client stubs
- Fully functional business objects on client side
- Complete EOF environment on client side
  - EOControl
  - EODistribution (instead of EOAccess)
  - EOInterface
  - EOApplication
  - EOGeneration

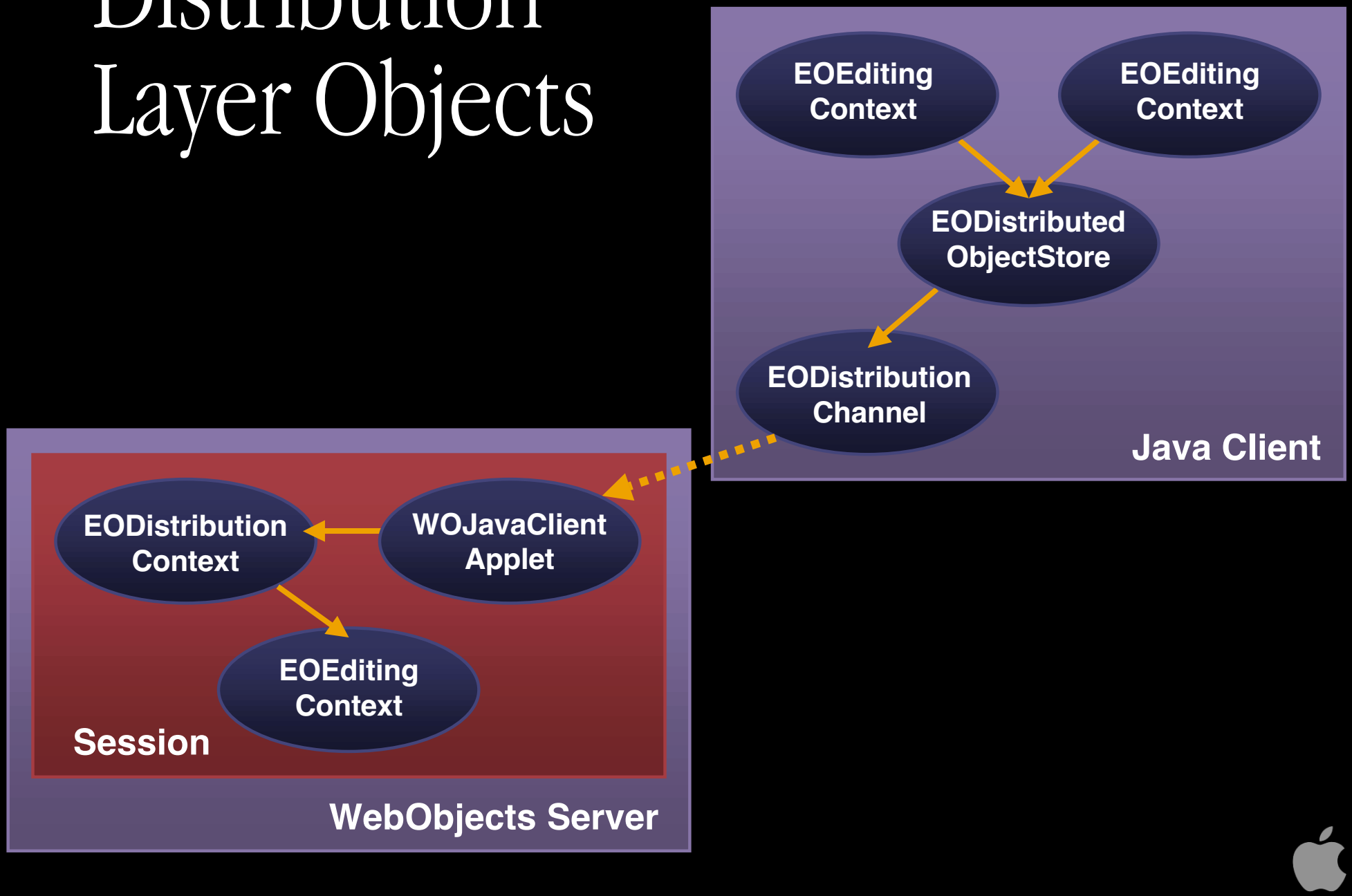
3-Tier Architecture



Distribution Layer Objects

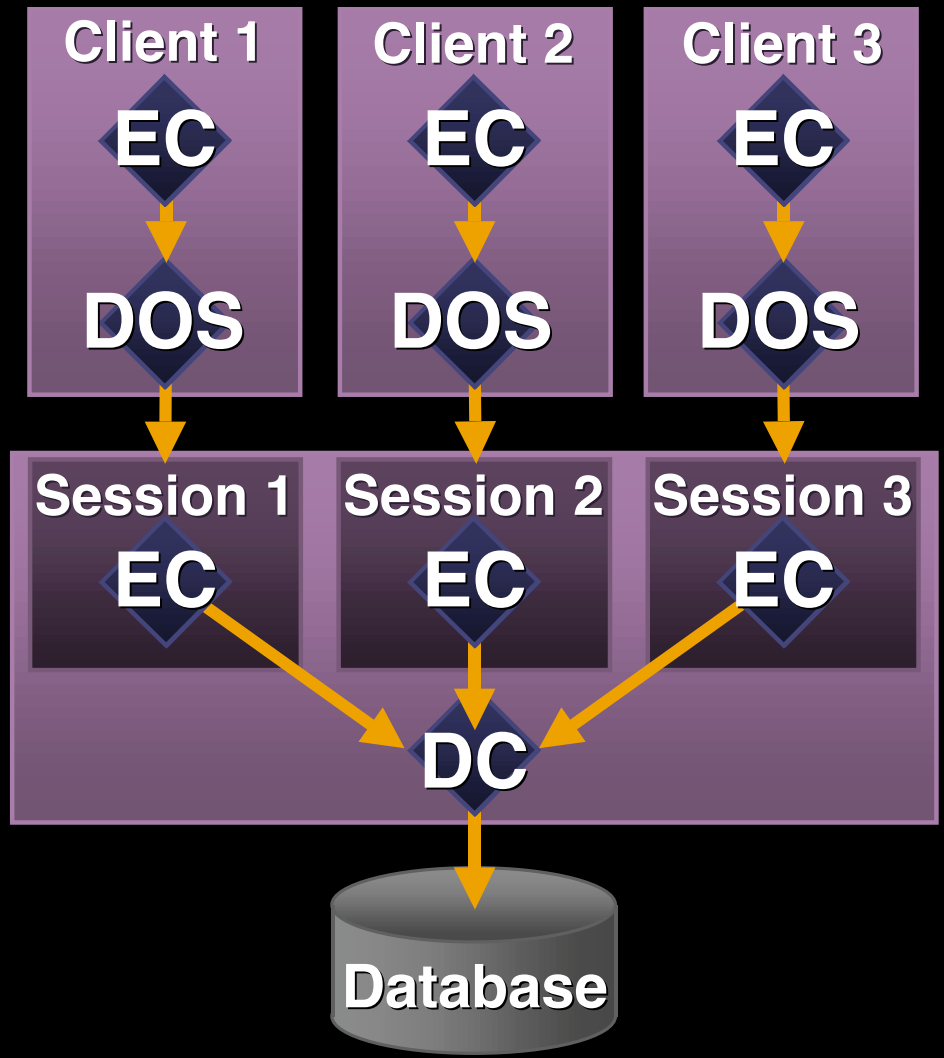
- Client side
  - EODistributedObjectStore
  - EODistributionChannel (EOHTTPChannel)
- Server side
  - WOJavaClientApplet (WOJavaClientComponent)
  - EODistributionContext

Distribution Layer Objects



Object Stores

- Editing contexts on the client behave like nested editing contexts to the session's editing context on the server
- Uni-directional connection (pull, no push)



Security

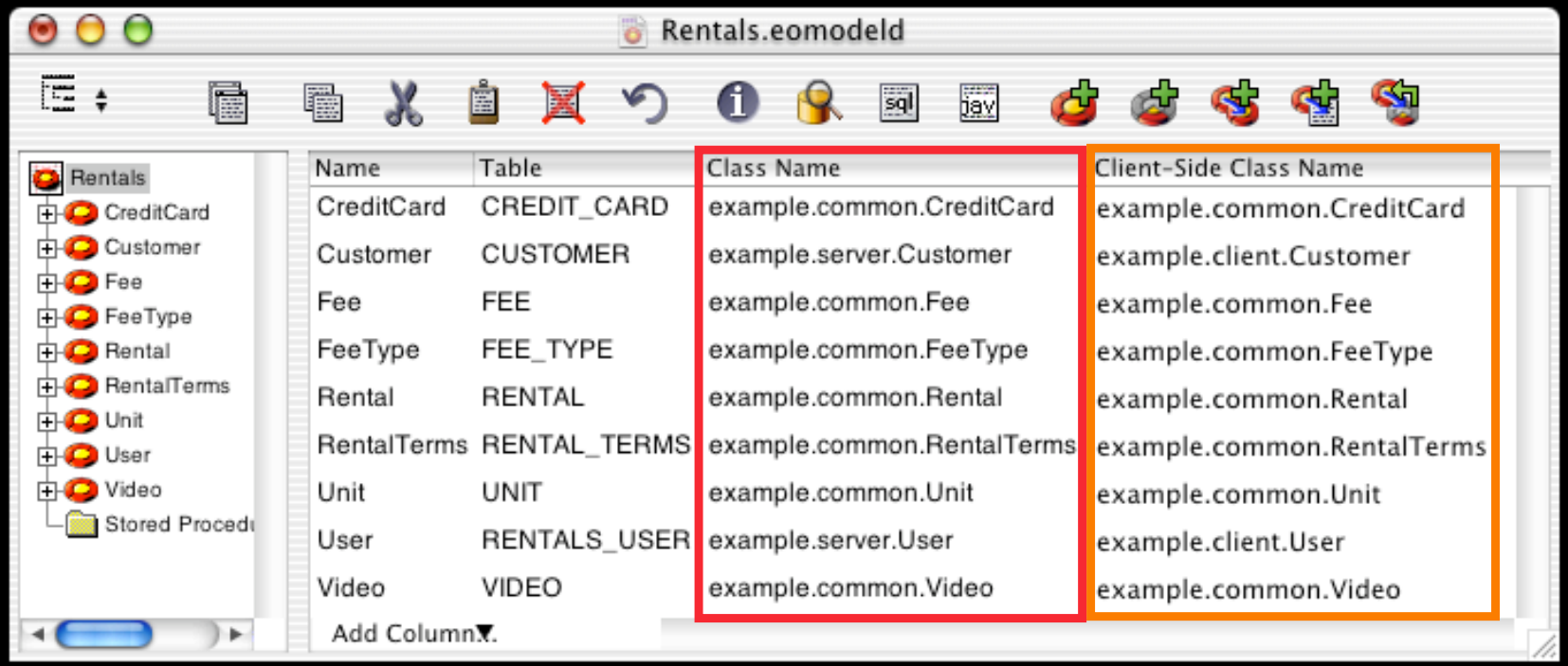
- Business logic partitioning
- Remote method invocations controlled by delegates on server
- Distribution Channels (SSL)

Business Logic Partitioning

- Business logic on client and server can be identical or client/server specific
  - Business objects can be represented by two different classes on client and server
  - Client side properties can be restricted (or extended with derived attributes)
- Also for performance optimization
  - Processing time
  - Transferred amount of data

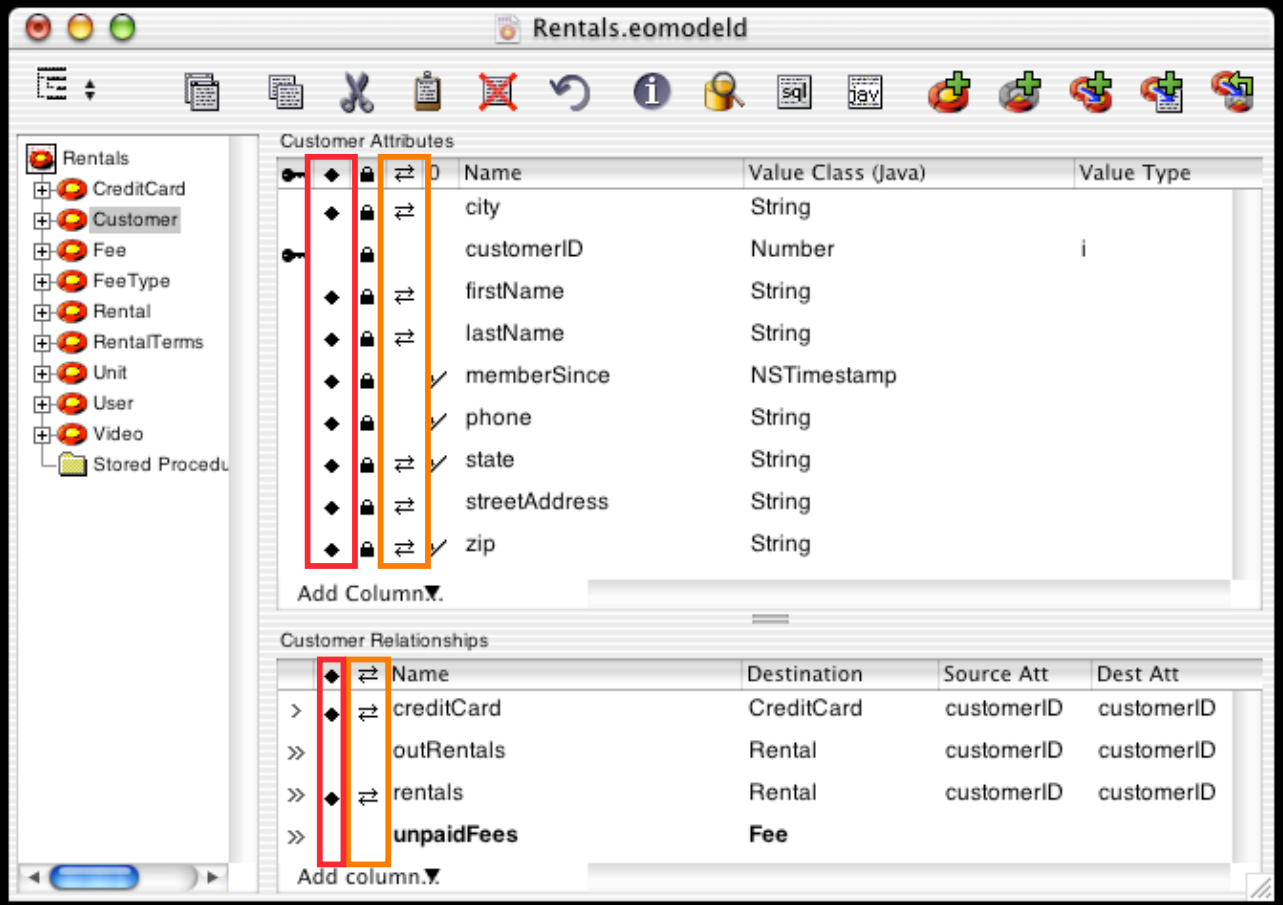
Business Logic Partitioning

- Different server side and client side classes



Business Logic Partitioning

- Different server side and client side properties



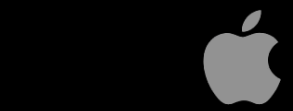
Business Logic Partitioning

- API on EOEnterpriseObject:

```
public Object invokeRemoteMethod (
    EOEditingContext editingContext ,
    String methodName ,
    Class[] argumentTypes ,
    Object[] arguments );

public void awakeFromClientUpdate (
    EOEditingContext editingContext
);

public void prepareValuesForClient ();
```

Business Logic Partitioning

- Naming convention: “clientSideRequest” for unrestricted invocations
- Class design recommendation
    - Common superclass
        - Shared functionality
        - Abstract method declarations for partitioned logic
    - Client subclass with remote method invocations
    - Server subclass with concrete method implementations



Business Logic Partitioning

```
package example.common;

import com.webobjects.eocontrol.*;

public abstract class Secret extends EOGenericRecord {

    // shared functionality
    // ...

    public abstract String secretValue ();
}
```

Business Logic Partitioning

```
package example.client;

import com.webobjects.eocontrol.*;

public class Secret extends example.common.Secret {

    public String secretValue () {
        return (String) invokeRemoteMethod(
            "clientSideRequestSecretValue",
            null, null);
    }
}
```

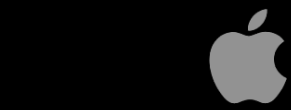
Business Logic Partitioning

```
package example.server;

import com.webobjects.eocontrol.*;

public class Secret extends example.common.Secret {

    public String secretValue () {
        return "secret";
    }

    public String clientSideRequestSecretValue() {
        return secretValue();
    }
}
```

Remote Method Invocations

- Invoke methods on the server related to application logic
    - Loading resources
    - User authentication
- API on EODistributedObjectStore
- EODistributedObjectStore is root object store:

```
EOEditingContext.defaultParentObjectStore();
```



Remote Method Invocations

- Stateful remote method invocation API:

```
public Object invokeRemoteMethodWithKeyPath (
        EOEditingContext editingContext ,
        String keyPath ,
        String methodName ,
        Class[] argumentTypes ,
        Object[] arguments ,
        boolean pushChanges );
```

Remote Method Invocations

- Stateless remote method invocation API:

```
public Object invokeStatelessRemoteMethodWithKeyPath (
        String keyPath ,
        String methodName ,
        Class[] argumentTypes ,
        Object[] arguments );
```

- No Enterprise Objects, but Global IDs okay
- Threadsafe
    - Great for background tasks

Remote Method Invocations

- Key path: Specifies the target of the method invocation relative to distribution context
- Example: “session”
- Special cases:
  - Null: Distribution context itself
  - Empty String: Invoke on distribution context targets
- Use distribution context notification to register additional targets:

  EORemoteMethodReceiverNeededNotification
- Naming convention: “clientSideRequest”

Remote Method Invocations

- Key path: Specifies the target of the method invocation relative to distribution context
- Example: “session”
- Special cases:
    - Null: Distribution context itself
    - Empty String: Invoke on distribution context targets
- Use distribution context notification to register additional targets:

    EORemoteMethodReceiverNeededNotification

- Naming convention: “clientSideRequest”

EODistributionContext Delegate

- Control access to business object fetching
  - Fetching with fetch specifications
  - Firing faults
- Restrict remote method invocations for each individual client
  - Key paths
  - Method names
- Set delegate when receiving notification:
  EODistributionContextInstantiatedNotification

Demo

Authentication in Java Client: DiscussionBoard

Norbert Schatz
WebObjects Engineering

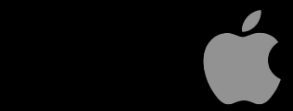
Summary

- Distribution layer is very efficient (copy distribution mechanism)
- Security
  - Business logic partitioning
  - Delegates to control remote method invocations
  - Distribution channel subclasses
- Web Start

WebObjects Lab

- Located downstairs in Room L
- Lab hours

Monday	12:00pm–6:00pm
Tuesday	9:00am–2:00pm
Wednesday	9:00am–6:00pm
Thursday	9:00am–6:00pm
Friday	9:00am–6:00pm



Roadmap

711 Advanced Data Modeling and Connectivity	Room A1 **Thurs., 3:30pm**
712 Advanced Enterprise Objects Frameworks	Room A1 **Thurs., 5:00pm**
FF013 WebObjects:	Room A1 **Fri., 3:30pm**

Who to Contact

Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

Bob Fraser
WebObjects Product Manager
webobjects@apple.com

Services Consulting, Integration, Training & Certification
(800) 848-6398
services@apple.com

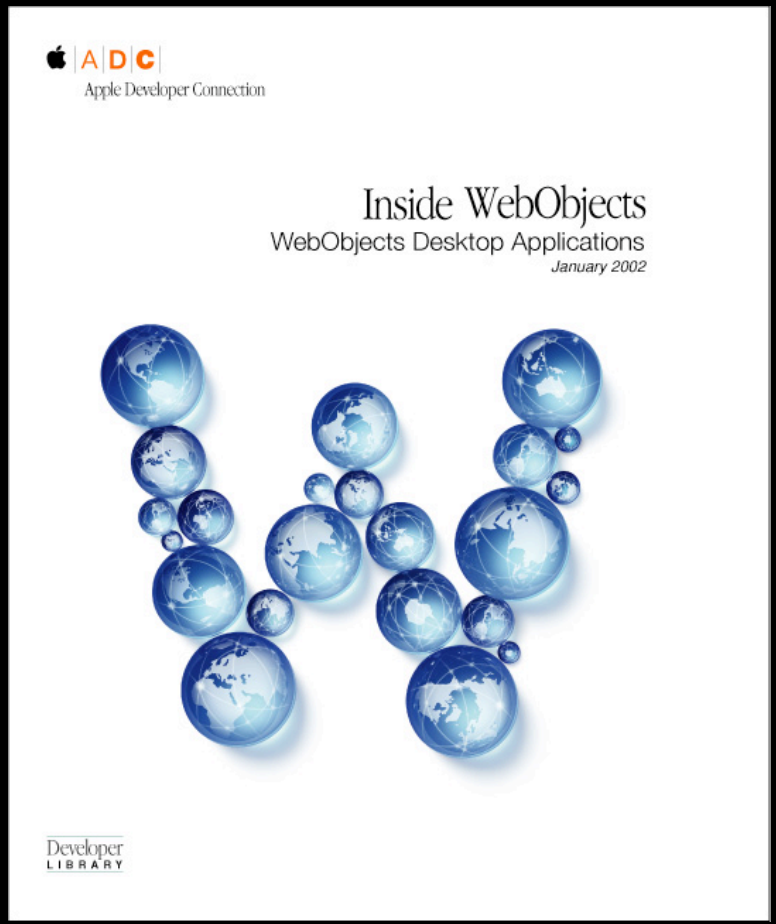
http://developer.apple.com/wwdc2002/urls.html

For More Information

- WebObjects Developer Documentation
  http://developer.apple.com/techpubs/webobjects
- Apple Professional Services Technical Support
  www.apple.com/services/technicalsupport
- Other places
  - **www.apple.com/webobjects**
  - **developer.apple.com/webobjects**
  - **www.apple.com/services**
  - **www.info.apple.com/webobjects**
- Subscribe to:
  - **webobjects-announce@apple.com**

Documentation

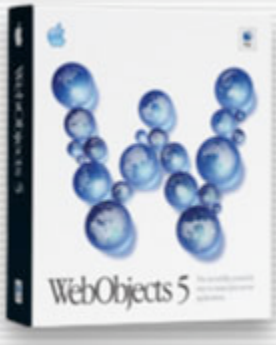
Java Client Desktop Applications



How to Access Documentation

- Most up-to-date: PDF and HTML
  http://developer.apple.com/techpubs/webobjects
- Hardcopy print-on-demand
  Vervante.com under Related Resources
- Product CD
  Documents folder and installed in
  /Developer/Documentation/WebObjects
- In the box (localized)
  Installation Guides, What's New, WebObjects Overview, Java Client Desktop Applications, Discovering WebObjects for HTML
- Check ADC News for latest updates
  http://developer.apple.com/devnews

Q&A



Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

http://developer.apple.com/wwdc2002/urls.html


WWDC2002


WWDC2002


WWDC2002